

## 発明の名称

画像処理装置および立体形状表示プログラム

## 発明の背景

### 発明の分野

#### 【0001】

本発明は、画像処理装置および立体形状表示プログラムに関し、より特定のには、特徴的なデータ構造を用いて立体形状を表示する画像処理装置、および当該装置で実行されるプログラムに関する。

### 背景技術の説明

#### 【0002】

周知のように、グラフィック分野においては、物体の移動や回転等の様々な表現が容易に可能になってきている。その中でも、ゲームの分野においては、キャラクターがゲーム実行のために移動可能な領域（以下、ゲームフィールドという）を、自由に移動できる表現が用いられてきている。

一般に、このゲームフィールドは、二次元平面で表され、平面上に形成される突起物等の物体は、平面座標の属性データとして高さ情報が与えられることによって表示される。キャラクターがゲームフィールドを移動する場合、移動先の平面座標の属性データを参照してキャラクターの高さ位置を制御することで、キャラクターが物体の表面に沿って移動できるようにしている。

#### 【0003】

ところで近年、ゲームフィールドを、例えば地球のような閉曲面の立体で表示して、キャラクターがこの立体表面上を縦横無尽に移動できるように表現することが望まれている。すなわち、ある地点から真っ直ぐに進み続けると、同一地点に再び戻ってくるようなゲームフィールド表現を可能にすることである。

しかしながら、上述した従来の画像処理技術では、ゲームフィールドを二次元平面で表示するため、三次元の立体を直接的に表示することができない。なお、簡単な立体（六面体等）については、複数の二次元平面を繋ぎ合わせて擬似的に

表示する方法も考えられるが、この方法の場合、キャラクタが移動するたびに全ての二次元平面の各データを対象に処理しなくてはならない。このため、結果を得るまでの処理時間が増大することに加え、処理時間にばらつきが生じるので、この方法は現実的ではない。

## 発明の概要

### 【0004】

それ故に、本発明の目的は、特徴的なデータ構造を用いて複雑な三次元の立体（ゲームフィールド、キャラクタ（敵キャラクタや建物等））を表示できると共に、キャラクタ等の特定点と立体面との位置関係を容易に把握することが可能な画像処理装置および当該装置で実行される立体形状表示プログラムを提供することである。

### 【0005】

本発明は、上記目的を達成するために、以下に述べる特徴を有している。

本発明の第1の局面は、複数の多角形で構成される所望の立体形状を表示するための画像データを出力する画像処理装置であって、記憶部、読出部、および画像データ出力部を備える。

記憶部は、空間上の任意の点を重心とし、各面が複数の多角形に分割された正多面体を規定し、当該重心から当該複数の多角形の各頂点に向かう方向を示すベクトルデータおよび当該重心と所望の立体形状の複数の多角形の各頂点との距離を設定するための距離データを記憶する。読出部は、記憶部からベクトルデータおよび距離データを読み出す。画像データ出力部は、重心から、読出部によって読み出されたベクトルデータに基づく方向で、距離データに基づく距離にある点を、所望の立体形状の複数の多角形の各頂点として、所望の立体形状を画像表示するための画像データを出力する。

### 【0006】

上記のように、第1の局面によれば、正多面体の複数の多角形の頂点毎にベクトルデータおよび距離データを持たせ、その距離データを任意に変更させることで、様々な立体形状を簡単に表示することができる。

#### 【0007】

ここで、正多面体の各面を分割している複数の多角形は、それぞれ大きさおよび形状が同じであることが好ましい。

このように、複数の多角形の大きさおよび形状を全て同じにすることで、データ管理が容易になる。

#### 【0008】

また、正多面体を正八面体にすることが好ましい。

このように、正多面体を正八面体にすれば、正多面体を展開させたときに正方形を形成することができるので、データ管理が容易になる。

#### 【0009】

さらに、好ましくは、他の物体の重心からの距離と、所望の立体形状上の接触処理対象となる多角形の重心からの距離とに基づいて、当該所望の立体形状と当該他の物体との接触判定を行う接触判定部をさらに備えるようにする。

このようにすれば、どんな複雑な立体形状であっても、特定の計算によってある空間座標値から処理対象となる多角形のデータを得ることができる。従って、ある空間座標に存在する物体と処理対象となる多角形との接触判定を容易に行うことが可能となる。

#### 【0010】

本発明の第2の局面は、複数の多角形で構成される所望の立体形状を表示するための画像データを出力するために、空間上の任意の点を重心とし、各面が複数の多角形に分割された正多面体を規定し、重心から複数の多角形の各頂点に向かう方向を示すベクトルデータおよび重心と所望の立体形状の複数の多角形の各頂点との距離を設定するための距離データを記憶する記憶部からデータ読み出し可能なコンピュータによって実行されるプログラムであって、読み出しステップ、および出力ステップを含む。

読み出しステップでは、記憶部からベクトルデータおよび距離データが読み出される。出力ステップでは、重心から、読み出されたベクトルデータに基づく方向で、距離データに基づく距離にある点を、所望の立体形状の複数の多角形の各頂点として、所望の立体形状を画像表示するための画像データが出力される。

#### 【0 0 1 1】

上記のように、第2の局面によれば、正多面体の複数の多角形の頂点毎にベクトルデータおよび距離データを持たせ、その距離データを任意に変更させることで、様々な立体形状を簡単に表示することができる。

#### 【0 0 1 2】

ここで、正多面体の各面を分割している複数の多角形は、それぞれ大きさおよび形状が同じであることが好ましい。

このように、複数の多角形の大きさおよび形状を全て同じにすることで、データ管理が容易になる。

#### 【0 0 1 3】

また、正多面体を正八面体にすることが好ましい。

このように、正多面体を正八面体にすれば、正多面体を展開させたときに正方形を形成することができるので、データ管理が容易になる。

#### 【0 0 1 4】

さらに、好ましくは、他の物体の重心からの距離と、所望の立体形状上の接触処理対象となる多角形の重心からの距離とに基づいて、当該所望の立体形状と当該他の物体との接触判定を行う接触判定ステップをさらに含むようにする。

この場合、接触判定ステップでは、他の物体の空間座標値の符号に基づいて、接触処理対象となる多角形が含まれる正多面体の面領域を選択させ、距離データから得られる平面方程式を用いて、選択された面領域の中から接触処理対象となる多角形を特定させ、さらに、特定された多角形の重心からの距離と、他の物体の重心からの距離とを比較し、当該他の物体と所望の立体形状との接触を判断させることが好ましい。

このようにすれば、どんな複雑な立体形状であっても、特定の計算によってある空間座標値から処理対象となる多角形のデータを得ることができる。従って、ある空間座標に存在する物体と処理対象となる多角形との接触判定を容易に行うことが可能となる。これにより、キャラクタが立体表面を連続して移動する等の場合であっても、高速に安定した処理を行うことが可能となる。また、複雑な形状との接触判定を行うソフトウェアの開発が容易に実現できる。

#### 【0015】

本発明の第3の局面は、コンピュータによって実行され、複数の多角形で構成される所望の立体形状を表示するためのプログラムおよびデータを記録する記録媒体であって、以下のデータ及びプログラムを記録する。

空間上の任意の点を重心とし、各面が複数の多角形に分割された正多面体を規定し、重心から当該複数の多角形の各頂点に向かう方向を示すベクトルデータ。重心と所望の立体形状の複数の多角形の各頂点との距離を設定するための距離データ。記憶部からベクトルデータおよび距離データを読み出すためのプログラム。および、重心から、読み出されたベクトルデータに基づく方向で、距離データに基づく距離にある点を、所望の立体形状の複数の多角形の各頂点として、所望の立体形状を画像表示するための画像データを出力するためのプログラム。

#### 【0016】

本発明のこれらおよび他の目的、特徴、局面、効果は、添付図面と照合して、以下の詳細な説明から一層明らかなになるであろう。

### 図面の簡単な説明

#### 【0017】

図1は、本発明の一実施形態に係る立体形状表示プログラムを実行する画像処理装置を含むビデオゲームシステムの構成例を示すブロック図である。

図2は、図1におけるメモリ43のメモリマップの一例を示す図である。

図3は、図1におけるゲームディスク47のメモリマップの一例を示す図である。

図4A及び図4Bは、本発明の一実施形態に係る立体形状表示プログラムで用いられる基本立体形状を示す図である。

図5A及び図5Bは、本発明の一実施形態に係る立体形状表示プログラムで用いられるデータ構造を説明するための図である。

図6Aは、本発明の一実施形態に係る立体形状表示プログラムで用いられるデータ構造を説明するための図である。

図6Bは、距離dの概念を説明するための図である。

図 7 は、本発明の一実施形態に係る立体形状表示プログラムによって表示可能な立体形状例を示す図である。

図 8 は、本発明の一実施形態に係る立体形状表示プログラムが、図 1 に示すビデオゲームシステム上で実行される手順を説明するためのフローチャートである。

図 9 は、任意の点と立体形状との位置関係を判断する場合の概念を説明する図である。

図 10 は、予め設定されるエリア番号の一例を説明する図である。

図 11 は、エリア番号を特定するための判断フローの一例を示す図である。

図 12 は、メモリ位置を特定するために予め有している対応表の一例を示す図である。

図 13 及び図 14 は、メモリ位置を特定するための処理フローの一例を示す図である。

## 好ましい実施例の説明

### 【0018】

本発明が提供する画像処理装置は、ポリゴンによる画像表示を必要とするグラフィックシステムやゲーム機器等に用いられ、それらを構成するCPUやメモリ等と協働して特徴的な処理を行うことで実現される。以下、本発明が提供する画像処理装置および当該装置で実行される立体形状表示プログラムを、図 1 ～図 14 を参照して説明する。

### 【0019】

(本発明を実現させるためのシステム環境)

図 1 に、本発明の立体形状表示プログラムを実行する画像処理装置を含むビデオゲームシステムの構成例を示す。図 1 において、ビデオゲームシステムは、メインプロセッサ 41、コプロセッサ 42、メモリ 43 およびドライブユニット 44 を備えるメインユニットと、ゲームコントローラ 46 と、表示部 45 と、ゲームディスク 47 とで構成される。

### 【0020】

コプロセッサ４２は、バス制御を行うためのバス制御回路と、ポリゴンの座標変換や陰影処理等を行うための信号プロセッサと、ポリゴンデータを表示すべき画像にラスタライズしかつフレームメモリに記憶可能なデータ形式（ドットデータ）に変換するための描画プロセッサとを含む（いずれも図示せず）。コプロセッサ４２には、ゲームディスク４７を駆動させるドライブユニット４４とメモリ４３とが接続される。また、コプロセッサ４２には、メインプロセッサ４１によって処理された音声信号および映像信号をそれぞれ出力するためのデジタル／アナログ変換器（図示せず）を介して、表示部４５が接続される。また、コプロセッサ４２には、ゲームコントローラ４６が接続される。

バス制御回路は、バスを介してメインプロセッサ４１からパラレル信号で与えられたコマンドをシリアル信号に変換して、ゲームコントローラ４６に供給する。また、バス制御回路は、ゲームコントローラ４６からシリアル信号で与えられたコマンドをパラレル信号に変換して、メインプロセッサ４１へ出力する。ゲームコントローラ４６から入力される操作状態を示すデータは、メインプロセッサ４１によって処理されたり、メモリ４３に一時記憶される等の処理が行われる。ゲームディスク４７に格納されているプログラムおよびメモリ４３に格納されているプログラムは、ゲームコントローラ４６から入力される操作に応じて、メインプロセッサ４１およびコプロセッサ４２で実行処理される。実行された結果は、コプロセッサ４２によって表示部４５の画面上に表示される。

#### 【００２１】

図１において、本発明の立体形状表示プログラムは、メモリ４３およびゲームディスク４７に格納され、コプロセッサ４２内で実行される。メモリ４３には、図２のメモリマップで示されるプログラムおよびデータが格納される。また、ゲームディスク４７は、ＤＶＤ等の記録媒体であって、図３のメモリマップで示されるプログラムおよびデータが格納される。

#### 【００２２】

メモリ４３において、プログラム領域には、ビデオゲームメインユニットを動作させるための基本的なシステムプログラムが格納されている。正八面体データ記憶領域には、後述する基本立体形状である正八面体のデータが格納される。計

算用バッファは、各種計算に用いられるバッファである。ポリゴンデータ領域には、1枚のフレーム画像を構成するために必要なポリゴンデータの情報が格納される。フレームバッファには、1フレームの画像が格納される。

ゲームディスク47において、メインプログラム領域には、ゲームを動作させる基本的なプログラムが格納されている。正八面体処理プログラム領域には、後述する正八面体を基本立体形状とする各処理を実行させるためのプログラムが格納されている。その他のプログラム領域には、メインプログラム以外のプログラムが格納されている。画像データ領域には、正八面体のデータや、ゲームのグラフィック表示に必要なポリゴンおよびテクスチャ等に関する様々なデータが格納されている。音データ領域には、ゲームの音表現に関する様々なデータが格納されている。

#### 【0023】

(格納される立体形状データの構造)

次に、図4～図7を用いて、本発明の一実施形態に係る立体形状表示プログラムで用いられるデータ構造を説明する。

#### 【0024】

本発明では、三次元空間の任意の点を原点とする特定の座標系において、重心Gを原点とする正八面体を、様々な立体形状を表示するための基本立体形状として用いる(図4A)。この特定の座標系は、原点(重心G)から正八面体の各頂点に向かって放射される放射線(原点から各頂点方向を示すベクトル)が座標軸として設定されており、この各々の座標軸は、その軸上にある正八面体の頂点の移動にそれぞれ用いられる。また、この正八面体の8つの平面は、それぞれ予め複数の三角形の領域に細分化されている。正八面体では、細分化される三角形が正三角形となる。この三角形の各頂点も、原点(重心G)から放射されるそれぞれの放射線(座標軸)上に位置するように設定されている。なお、図4Aの例では、1つの平面の各辺をそれぞれ4等分する16個の三角形に分割しているが、他の数に分割してもかまわない。

なお、特定の座標系の原点がゲームフィールド(直交座標系)の原点と異なる場合、特定の座標系の原点をゲームフィールドの原点から相対的に判断する必要



がある。

#### 【0025】

正八面体の各平面を分割している三角形には、それぞれ固有の識別番号が設定される。この識別番号は、次のようにして設定される。

まず、正八面体のY座標が負値である4つの平面を正值側へそれぞれ展開し（図4Bの実線矢印）、正八面体の8つの平面を上方向（同図の白抜き矢印方向）から見た二次元平面（図5A）で表す。そして、この二次元平面で表された各三角形（直角二等辺三角形となる）に対して、図5Bのように左上から右下に向かって順番に識別番号（0～127）を与える。この識別番号は、メモリ等の記憶装置内のアドレスを特定できる情報である。この二次元平面では、縦横の線で囲まれるマス目がXZ座標の値によって区切られ、マス目内の斜線がY座標の値によって区切られていることが分かる。一方、メモリ内には、予め識別番号毎にデータ格納領域が確保されており（図6A）、そこには原点（重心G）から三角形の各頂点までの距離d（図6B）、三角形の平面方程式および表面の模様等のデータが格納されている。なお、三角形の平面方程式は、距離dから求めることも可能である。距離dから三角形の平面方程式を求めるようにすれば、格納するデータ量が少なくて済み、距離dの大きさを自在に変化させる場合に有利となる。

これにより、正八面体を構成する各三角形に関するメモリ内のデータを、識別番号を介して参照することが可能となる。

#### 【0026】

このように、本発明では、基本となる正八面体を構成する三角形毎に、その三角形を表示するためのデータがそれぞれ格納されているので、このデータを任意に変更することで様々な形状の立体を簡単に表示することができる。すなわち、本発明では、平面の各頂点が重心からの距離dで与えられる特定の座標系を用いて、立体形状を表示している。このため、正八面体を構成する三角形の各頂点を、座標軸上を距離dに従ってそれぞれ移動させるだけで、正八面体を簡単に変形することができるので、様々な立体表示が容易に可能になるのである。例えば、原点から全ての三角形の各頂点までの距離を同じにすれば、球体に近い立体を表示することができる（図7）。

## 【0027】

（立体形状表示プログラムの実行手順）

次に、図8を参照して、上述したデータ構造を用いた立体形状表示プログラムが、図1に示すビデオゲームシステム上でどのように実行され、作成する立体（作成立体）がディスプレイの画面に表示されるのかを説明する。

## 【0028】

始めに、電源投入時に動作するプログラムによって初期設定が行われる（ステップS11）。この初期設定では、ゲームディスク47がセットされているか否かやゲームコントローラ46が接続されているか否か等が検出される。初期設定が完了すると、ゲームディスク47からメモリ43へ必要なデータが転送される（ステップS12）。次に、メインプロセッサ41が、メモリ43のプログラム領域に記憶されている立体作成プログラムの実行を開始する（ステップS13）。続いて、メインプロセッサ41が、メモリ43の正八面体データ記憶領域から、基本立体の重心座標、基本三角形の各頂点座標および距離データを読み出し（ステップS14）、これらの座標およびデータから作成立体の各頂点座標を計算する（ステップS15）。そして、メインプロセッサ41が、作成立体の各頂点座標を、メモリ43のポリゴンデータ領域に転送する（ステップS16）。

この後、ゲームシステムでは、作成立体以外のその他のキャラクタの表示処理や（ステップS17）、後述する作成立体とその他のキャラクタとの接触判定処理や（ステップS18）、キャラクタの移動や攻撃処理等のその他のゲームに関する処理が（ステップS19）、必要に応じて行われる。

次に、コプロセッサ42が、メモリ43に記憶された各頂点座標に基づいてポリゴンを作成し、そのポリゴンにテクスチャを貼り付けることにより表示部45に表示するための画像データを作成し、その画像データをメモリ43のフレームバッファに記憶させる（ステップS20）。最後に、コプロセッサ42が、フレームバッファの画像データに基づいて、表示部45に画像信号を出力する（ステップS21）。上記ステップS13～S21の処理は、ゲームが終了するまで繰り返して行われる（ステップS22）。

## 【0029】

（任意の点と立体形状との位置関係判断）

次に、図9～図14を用いて、上述した立体形状データで表示される立体Mと空間のある任意の点P（ $p_x$ ， $p_y$ ， $p_z$ ）との、位置関係を判断する手法を説明する。この立体Mと点Pの関係としては、例えば、立体M＝地面、点P＝キャラクターの足の裏、立体M＝ビル、点P＝キャラクターのお腹、立体M＝サーカスのオートバイがその中を走行する球体、点P＝オートバイ、等が考えられる。

#### 【0030】

上述した立体M平面上の細分化された三角形の内、空間の点Pと位置関係を判断すべき三角形は、点Pを原点に向かって正八面体上に投影した点Qの位置から特定される（図9）。そして、その特定された三角形の現在の位置データから、立体M表面の平面方程式を求めることができる。そこで、本発明では、以下のよう

#### 【0031】

最初に、基本立体形状である正八面体の8つの平面のどの平面において、点Pとの位置関係判断処理を行うべきかを決定する。

この決定は、次のようにして行われる。まず、正八面体の8つの平面には、図10に示すようにエリア番号（0～7）がそれぞれ設定される。図10の例では、X座標値の正負でエリア番号の値が1つ、Y座標値の正負でエリア番号の値が2つ、およびZ座標値の正負でエリア番号の値が4つ変化するルールで各エリア番号が付されている。従って、図11に示す判定フローを実行し、点PのXYZ座標値の各符号を用いて処理対象平面のエリア番号 $a_r$ を決定する。例えば（ $p_x$ ， $p_y$ ， $p_z$ ）＝（10，15，－8）のようにZ値だけが負である点Pの場合には、まず $p_x < 0$ を満たさないので（ステップS71，No）ステップS72が実行されて $a_r = 1$ となり、次に $p_y < 0$ を満たさないので（ステップS74，No）ステップS75が実行されて $a_r = 3$ （＝1＋2）となり、最後に $p_z < 0$ を満たすので（ステップS77，Yes）ステップS79が実行されて、エリア番号 $a_r = 3$ が決定される。なお、この判定フローと同時に、点P（ $p_x$ ， $p_y$ ， $p_z$ ）の絶対値（ $a_x$ ， $a_y$ ， $a_z$ ）が求められる。

#### 【0032】

次に、この決定されたエリア番号  $a_r$  に従って、処理対象平面のデータ格納メモリ先頭位置  $m_s$ （エリア内でY軸に接する三角形の識別番号）、点QのX座標変化に対するメモリ上での位置変化割合  $v_x$ （エリア内で先頭位置  $m_s$  のマス目がX軸方向に1つ移動したときの識別番号の変化量）、点QのZ座標変化に対するメモリ上での位置変化割合  $v_z$ （エリア内で先頭位置  $m_s$  のマス目がZ軸方向に1つ移動したときの識別番号の変化量）、およびY座標によるマス目内の三角形選択ベクトル  $v_y$ （マス目内の三角形を移動したときの識別番号の変化量）が決定される。これらの値は、設定された識別番号およびエリア番号  $a_r$  に基づいて定義される対応表を参照して決定される。

例えば、図5Bおよび図10に示した例の場合には、図12の対応表が定義されることとなる。図12のエリア番号3で説明すると、メモリ先頭位置  $m_s$  は、Y軸に接する三角形の識別番号「56」となる。位置変化割合  $v_x$  は、マス目をX軸の右方向（プラス方向）に1つ移動したとき変化量「2（＝58－56）」となる。位置変化割合  $v_z$  は、マス目をZ軸の上方向（マイナス方向）に1つ移動したとき変化量「－16（＝40－56）」となる。三角形選択ベクトル  $v_y$  は、マス目内の三角形を移動する変化量「1（＝57－56）」となる。なお、この対応表の内、エリア番号0，1，4および5関する値は、図5Aのように二次元平面の展開した際に、X軸およびZ軸の向きが入れ替わっているため、位置変化割合が  $v_x$  と  $v_z$  とで逆になっている。また、後述する計算処理では座標の絶対値が使用されるため、図12に対応表の各符号は、求められる点Qの座標値が絶対値であることを前提に設定されている。

#### 【0033】

次に、点Qの座標（ $q_x$ ， $q_y$ ， $q_z$ ）を求める。

点Qの座標は、点Pと原点とを結ぶ直線と、正八面体の処理対象平面との、交点の方程式を解くことで求めることができる。なお、点Pの座標値は、その符号によって対象となる平面（エリア番号  $a_r$ ）が決定されているので、下記の計算には絶対値（ $a_x$ ， $a_y$ ， $a_z$ ）を用いればよい。従って、点Qの座標は、上記直線と処理対象平面との交点座標の絶対値で与えられることとなる。また、結果としてマス目単位での値を必要とするので、小数点以下は切り捨てる。この処理

をフローで表すと、図13のようになる。なお、変数nは、平面の各辺を等分する数であり、この例では「4」となる。

例えば、絶対値  $(a_x, a_y, a_z) = (10, 20, 15)$  である場合、

$$f = 4 / (10 + 20 + 15) = 4 / 45$$

となる。従って、点Qの各座標値は、

$$q_x = rd[f * 10] = rd[0.88\cdots] = 0$$

$$q_y = rd[f * 20] = rd[1.77\cdots] = 1$$

$$q_z = rd[f * 15] = rd[1.33\cdots] = 1$$

となる。なお、 $rd[X]$  は、Xの小数点以下を切り捨てる関数である。

【0034】

最後に、求められたメモリ先頭位置ms、X座標位置変化割合vx、Z座標位置変化割合vz、三角形選択ベクトルvy、および点Qの座標から、点Pが投影される三角形の識別番号、すなわち三角形のデータが格納されたメモリ位置mmを次のように計算する。

図14を参照して、まず、点QのXZ座標にX座標位置変化割合vxおよびZ座標位置変化割合vzをそれぞれ掛けた値を、メモリ先頭位置msに足し合わせることで、マス目単位での目的のメモリ位置が求められる（ステップS101）。そして、所定の条件に従って（ステップS102）、マス目内のどちらの三角形が該当するかが選択されて、メモリ位置mmが決定される（ステップS103）。

例えば、上述した例の変数を全て適応させると、

$$mm = 56 + 0 * 2 + 1 * (-16) = 40$$

$$q_x + q_y + q_z \neq n - 1$$

$$mm = 40 + 1 = 41$$

となり、メモリ位置 $mm = 41$ を求めることができる。

【0035】

そして、メモリ位置mmが求まると、そこから三角形に関するデータを取得して、点Pとの位置関係や接触判定を行うことができる。例えば、取得した三角形の表面方程式と、キャラクタの基準点（足の裏等）の座標と原点座標とを結ぶ直

線との交点を、数学的に求めて接触判定を行うことができる。原点から交点までの距離が、原点から足下の座標までの距離より長ければ、キャラクタが立体Mと接触していると判断できる。また、取得した三角形に関するデータを書き換えて、立体Mの形状を自由に変化させることができる。

#### 【0036】

以上のように、本発明の一実施形態に係る画像処理装置および立体形状表示プログラムによれば、立体の形状がどんなに複雑になろうとも、特定の計算により空間の座標値から参照すべき立体の表面データが得られる。これにより、キャラクタが立体表面を連続して移動する等の場合であっても、高速に安定した処理を行うことが可能となる。また、複雑な形状との接触判定を行うソフトウェアの開発が容易に実現できる。

#### 【0037】

なお、上記実施形態では、基本立体形状が正八面体である場合を説明したが、計算処理がやや複雑になっても構わなければ、他の正多面体を用いて同様に実現することは可能である。

また、上述した基本立体形状やそれを変形させた立体形状は、計算に用いられるためだけの図形であって、実際にポリゴンで形作られない。画面表示されるのは、これらの立体形状に基づいて生成されるポリゴンにテクスチャが貼り付けられた画像である。

#### 【0038】

以上、本発明を詳細に説明してきたが、前述の説明はあらゆる点において本発明の例示に過ぎず、その範囲を限定しようとするものではない。本発明の範囲を逸脱することなく種々の改良や変形を行うことができることは、言うまでもない。

## クレーム

1. 複数の多角形で構成される所望の立体形状を表示するための画像データを出力する画像処理装置であって、

空間上の任意の点を重心とし、各面が複数の多角形に分割された正多面体を規定し、当該重心から当該複数の多角形の各頂点に向かう方向を示すベクトルデータおよび当該重心と前記所望の立体形状の複数の多角形の各頂点との距離を設定するための距離データを記憶する記憶手段、

前記記憶手段からベクトルデータおよび距離データを読み出す読出手段、および

前記重心から、前記読出手段によって読み出された前記ベクトルデータに基づく方向で、前記距離データに基づく距離にある点を、前記所望の立体形状の複数の多角形の各頂点として、前記所望の立体形状を画像表示するための画像データを出力する画像データ出力手段を備える。

2. クレーム1に従属する画像処理装置であって、

前記正多面体の各面を分割している複数の多角形は、それぞれ大きさおよび形状が同じであることを特徴とする。

3. クレーム1に従属する画像処理装置であって、

前記正多面体が、正八面体であることを特徴とする。

4. クレーム1に従属する画像処理装置であって、

他の物体の前記重心からの距離と、前記所望の立体形状上の接触処理対象となる多角形の前記重心からの距離とに基づいて、当該所望の立体形状と当該他の物体との接触判定を行う接触判定手段をさらに備える。

5. 複数の多角形で構成される所望の立体形状を表示するための画像データを出力するために、空間上の任意の点を重心とし、各面が複数の多角形に分割された正多面体を規定し、重心から複数の多角形の各頂点に向かう方向を示すベクトルデータおよび重心と所望の立体形状の複数の多角形の各頂点との距離を設定するための距離データを記憶する記憶手段からデータ読み出し可能なコンピュータによって実行されるプログラムであって、

前記記憶手段から前記ベクトルデータおよび前記距離データを読み出すステッ

プ、および

前記重心から、読み出された前記ベクトルデータに基づく方向で、前記距離データに基づく距離にある点を、前記所望の立体形状の複数の多角形の各頂点として、前記所望の立体形状を画像表示するための画像データを出力するステップを含む。

6. クレーム5に従属するプログラムであって、

前記正多面体の各面を分割している複数の多角形は、それぞれ大きさおよび形状が同じであることを特徴とする。

7. クレーム5に従属するプログラムであって、

前記正多面体が、正八面体であることを特徴とする。

8. クレーム5に従属するプログラムであって、

他の物体の前記重心からの距離と、前記所望の立体形状上の接触処理対象となる多角形の前記重心からの距離とに基づいて、当該所望の立体形状と当該他の物体との接触判定を行うステップをさらに含む。

9. クレーム8に従属するプログラムであって、

前記接触判定を行うステップは、

前記他の物体の空間座標値の符号に基づいて、接触処理対象となる多角形が含まれる前記正多面体の面領域を選択するステップ、

前記距離データから得られる平面方程式を用いて、選択された前記面領域の中から前記接触処理対象となる多角形を特定するステップ、および

前記特定された多角形の前記重心からの距離と、前記他の物体の前記重心からの距離とを比較し、当該他の物体と前記所望の立体形状との接触を判断するステップからなる。

10. コンピュータによって実行され、複数の多角形で構成される所望の立体形状を表示するためのプログラムおよびデータを記録する記録媒体であって、

空間上の任意の点を重心とし、各面が複数の多角形に分割された正多面体を規定し、重心から当該複数の多角形の各頂点に向かう方向を示すベクトルデータ、

前記重心と前記所望の立体形状の複数の多角形の各頂点との距離を設定するための距離データ、



前記記憶手段から前記ベクトルデータおよび前記距離データを読み出すためのプログラム、および

前記重心から、読み出された前記ベクトルデータに基づく方向で、前記距離データに基づく距離にある点を、前記所望の立体形状の複数の多角形の各頂点として、前記所望の立体形状を画像表示するための画像データを出力するためのプログラムを記録する。

## 開示の概要

特徴的なデータ構造を用いて複雑な三次元の立体を表示できると共に、キャラクター等の特定点と立体面との位置関係を容易に把握することが可能な立体形状表示プログラムおよび画像処理装置を提供する。各頂点が三次元軸上のいずれかに存在する正八面体を基本立体形状として用いる。正八面体の8つの平面は、それぞれ予め複数の三角形領域に細分化されており、メモリ内のアドレスを特定できる固有の識別番号が付される。メモリには、予め識別番号毎にデータ格納領域が確保されており、原点から三角形の各頂点までの距離、三角形の平面方程式および表面の模様等のデータが格納されている。このように、基本となる正八面体を構成する三角形毎に、その三角形を表示するデータが各々格納されているので、識別番号を介して容易にデータ参照が可能であり、任意のデータ変更によって様々な立体形状を簡単に表示可能である。